PATENT
Attorney Docket No. YOR920030279US1

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of

Sophia KRASIKOV et al.

Serial No: 10/684,552

Filed: October 14, 2003

For: AGGREGATION OF DOCUMENT ELEMENTS INTO RUNTIME CODE

Examiner: Wei, Zheng

Art Unit: 2192

SUPPLEMENTAL APPEAL BRIEF

Board of Patent Appeals and Interferences
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450

Dear Sir:

The Appellant submits this brief pursuant to 37 C.F.R. §41.37(a)(1) in furtherance of the Notice of Appeal filed March 14, 2008 and the Notice of Non-Compliant Appeal Brief dated March 11, 2009, setting a one-month shortened statutory period of reply expiring April 13, 2009.

No fee is believed due with this Appeal Brief, however, should another fee be required please charge Deposit Account 50-0510.

Real Party in Interest

The real party in interest is International Business Machines Corporation.

Related Appeals and Interferences

The Appellants' legal representative does not know of any other appeal, interference or judicial proceeding which will affect or be directly affected by or have bearing on the Board's decision in the pending appeal.

Status of Claims

Claims 1-3, 5-9, 14-19 and 21-31 are pending in the present Application, with claims 1, 14, 17 and 26 being independent claims. Claims 4, 10-13 and 20 are cancelled. The final rejection of claims 1-3, 5-9, 14-19 and 21-31 is appealed.

Status of Amendments

No amendments to the claims were made after the Final Office Action dated December 14, 2007 ("FOA").

Summary of the Claimed Subject Matter

Independent claim 1 recites a method for creating a minimum executable unit. Application, pp. 7, ll. 23-24, Fig. 2, item 206. As used in the Application, a "minimum executable unit" is a self-contained executable file comprising at least the minimum executable code and documentation for an application. Application, pp. 7, ll. 24-25.

The method includes a receiving operation for receiving at least one block of source code. Application, pp. 7, ll. 23-24, Fig. 2, item 110. Another receiving operation receives at least one document element associated with the source code. Application, pp. 10, ll. 20-21, Fig. 2, item 112.

A demarcating operation demarcates a target audience of the document element. Application, pp. 9, ll. 15-16, pp. 10, Table 1, Fig. 3, item 310. The method further includes validating consistency between the source code and the document element. Application, pp. 11, ll. 24-26, Fig. 4, item 416. At a building operation, an executable unit is built, with the executable unit being responsive to both the source code and the document element. Application, pp. 6, ll. 18-27, Fig. 2, item 206.

Independent claim 14 recites a system for creating a minimum executable unit. Application, pp. 7, ll. 23-24, Fig. 2, item 206. The system includes at least one block of source code. Application, pp. 7, ll. 23-24, Fig. 2, item 110. At least one document element is associated with the source code. Application, pp. 10, ll. 20-21, Fig. 2, item 112. The document element includes an indicia specifying a human audience that the document element is targeted to. Application, pp. 9, ll. 15-16, pp. 10, Table 1, Fig. 3, item 310. A validation criteria is configured to validate the consistency between the source code and the document element. Application, pp. 11, ll. 24-26, Fig. 4, item 416. A builder is configured to construct an executable unit, the executable unit including the document element embedded in the source code. Application, pp. 6, ll. 18-27, Fig. 2, item 206.

Independent claim 17 recites a computer program product embodied in a tangible media. Application, pp. 4, ll. 29-30, Fig. 2, item 204. The computer program product includes computer readable program codes coupled to the tangible media for creating a minimum executable unit. Application, pp. 7, ll. 23-24, Fig. 2, item 206. The computer readable program codes are configured to cause the program to receive at least one block of source code. Application, pp. 7, ll. 23-24, Fig. 2, item 110. The computer readable program codes are also configured to cause the program to receive at document element associated with the source code. Application, pp. 10, ll. 20-21, Fig. 2, item 112. The document element includes an indicia specifying a human audience that the document element is targeted to. Application, pp. 9, ll. 15-16, pp. 10, Table 1, Fig. 3, item 310. A validating operation validates the consistency between the source code and the document element. Application, pp. 11, ll. 24-26, Fig. 4, item 416. At a building operation, an executable unit is built, with the executable unit being responsive to both the source code and the document element. Application, pp. 6, ll. 18-27, Fig. 2, item 206.

Independent claim 26 recites program code embodied on computer-readable media. Application, pp. 4, ll. 29-30, Fig. 2, item 204. The program code includes at least one block of source code providing executable computer instructions. Application, pp. 6, ll. 18-21, Fig. 1, item 110. At least one documentation string is coupled to the source code. Application, pp. 10, ll. 20-21, Fig. 2, item 112. The document element includes a method indicia identifying the method where the documentation string is located. Application, pp. 10, Table 1. A target indicia is configured to specify a human audience that the document element is targeted to. Application, pp. 9, ll. 15-16, pp. 10, Table 1, Fig. 3, item 310. A granularity indicia is configured to identify what document the documentation string should be a part of. Application, pp. 10, Table 1, Fig. 3, item 312.

Grounds for Rejection to be Reviewed on Appeal

I. Claims 26-29 are rejected under 35 U.S.C. § 112, first paragraph, as failing to comply with the enablement requirement.

II. Claims 26-29 are rejected under 35 U.S.C. § 101 as directed to non-statutory subject matter.

III. Claims 1-8 and 14-25 are rejected under 35 U.S.C. § 103(a) as obvious over Patent No. 6,252,597 issued to Crockett ("Crockett") in view of "WinZip(R) Self-Extractor Personal Edition", pp. 1-7 ("WinZip").

IV. Claims 9 and 25-29 are rejected under 35 U.S.C. § 103(a) as obvious over Crockett in view of WinZip and "FCC Computer Security Notice", pp. 1-2 ("FCC").

V. Claims 30 and 31 are rejected under 35 U.S.C. § 103(a) as obvious over Crockett in view of WinZip and Dirnitri van Heesch, "doxygen Maunal for version 1.2.10", pp. 1-19 ("Heesch").

Argument

I. CLAIMS 26-29 ARE ENABLED UNDER 35 U.S.C. § 112, FIRST PARAGRAPH

Claim 26 recites, in part, "at least one block of source code providing executable computer instructions." In rejecting claim 26 under 35 U.S.C. § 112, the Examiner argues that source code "itself cannot provide 'executable computer instructions' without passing compilation or interpretation and generating executable code phases as being well known in the art." FOA, pp. 4-5. The Appellants respectfully disagree.

It is well known to those skilled in the art that source code may be compiled or interpreted. "In interpreter languages, an executable program can be source code in the proper format." Microsoft Computer Dictionary Fifth Edition, Microsoft Press , pp. 200 (2002) (definition for executable program) (emphasis added). Contrary to the Examiner's position, the concept of a "block of source code providing executable computer instructions" is well within the grasp of those skilled in the art. Thus, the rejection of claim 26 under 35 U.S.C. § 112 is in error.

In section 10A of the Examiner's Answer dated July 22, 2008 ("EA"), the Examiner basically contends that "source code itself does not directly provide executable instructions." EA, pp. 14 (emphasis in original). This argument should not be given any weight by the honorable Board.

First, the Examiner provides no evidence supporting the opinion that interpreted source code does not provide computer executable instructions. The Examiner's Answer makes no finding of fact and cites no evidence in making this assertion. By contrast, the Appellant has provided evidence contradicting the Examiner's position. As described in Exhibit A of the Appeal Brief, "an executable program can be source code."

Second, the claims at issue do not recite that the source code directly provide executable instructions. The requirement that the source code "directly" provide executable instructions is a misreading of the claim. The Examiner's assertions about source code "directly" providing executable instructions are a red-herring; arguments against claim limitations that do not exist in the application.

Third, even if "source code providing executable computer instructions" must first be interpreted, this argument does not address the legal issue of whether one skilled in the art would be capable of making and using the claimed invention. The Appellant has introduced ample evidence in the record that the concept of executable source code is well understood by one of skill in the art. The Examiner offers no evidence why one skilled in the art would not be capable of making and using executable source code.

Claims 27-29 are dependent on and further limit claim 26. Since the rejection of claim 26 is believed in error, the rejection of claim 27-29 is also believed in error for at least the same reasons as claim 26.

The Appellants submit the rejection of claims 26-29 is in error and respectfully request that the rejection of claims 26-29 under 35 U.S.C. § 112 be reversed by the honorable Board.

II. CLAIMS 26-29 RECITE STATUTORY SUBJECT MATTER

Claim 26 was rejected under 35 U.S.C. § 101 as allegedly directed to non-statutory subject matter. The Final Office Action observes claim 26 recites "program code embodied on computer-readable media." However, the Examiner argues "said program code does not appear to produce/perform any functions so as to satisfy the practical application requirement. Therefore it is directed to nonfunctional descriptive material." FOA, pp. 6. The Appellants respectfully disagree.

As noted above, claim 26 recites "at least one block of source code providing executable computer instructions." Executable computer instructions cause computer processors to perform logic and arithmetic operations. Thus, the Appellants submit claim 26 is directed to functional material rather than descriptive material.

When functional descriptive material is recorded on some computer-readable medium it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive material to be realized. MPEP 2106. As noted by the Examiner, claim 26 recites, "program code embodied on computer-readable media."

The Examiner also argues in section 10B of the Examiner's Answer that the claim does not specify steps/activities/acts that the executable instructions are required to perform. EA, pp. 15. In this regard, the Examiner fails to consider the claim as a whole. The exclusive focus on the source code element of the claim is in error and leads the Examiner to the erroneous conclusion that the entire claim is non-functional.

The Appellants therefore submit claim 26 is directed to functional material recorded on computer-readable medium, and is statutory subject matter under 35 U.S.C. § 101.

Claims 27-29 are dependent on and further limit claim 26. Since the rejection of claim 26 is believed in error, the rejection of claim 27-29 is also believed in error for at least the same reasons as claim 26.

The Appellants submit the rejection of claims 26-29 is in error and respectfully request that the rejection of claims 26-29 under 35 U.S.C. § 101 be reversed by the honorable Board.

III. CLAIMS 1-8 AND 14-25 ARE NOT OBVIOUS OVER CROCKETT IN VIEW OF WINZIP

To establish a *prima facie* case of obviousness under 35 U.S.C. § 103, the prior art references must teach or suggest all the claim limitations. *See* MPEP 2143 et seq. It is well settled that "rejections on obviousness grounds cannot be sustained by mere conclusory statements; instead, there must be some articulated reasoning with some rational underpinning to support the legal conclusion of obviousness." In re Kahn, 441 F.3d 977, 988, 78 USPQ2d 1329, 1336, quoted with approval in KSR Int'l Co. v. Teleflex Inc., 127 S. Ct. 1727, 1741, 82 USPQ2d 1385, 1396 (2007).

Claim 1

Claim 1 recites, in part, "demarcating a target audience of the document element." In rejecting claim 1, the Office Action points to Figure 6, elements 64 and 68 and related text as disclosing a demarcation of a target audience of a document element. FOA, pp. 7. The Appellant disagree.

The Examiner argues, "Figure 6, element 68 'Documentation

Format Information and its tags element 64, for example tag or element 'Job Name', 'Job Step Name' are used to identify/indicate 'an element to be included in the documentation' and element 66 Text which are used to identified the text corresponds with the element for the given format. Therefore, these elements/tags/texts demarcate/identify what elements need to be documented in the documentation." FOA, pp. 1.

The Appellants note the Examiner does not provide any rational why element 64 or element 66 of Crockett is regarded as demarcating a target audience, as required by claim 1. The Examiner simply states these elements are used to identify text corresponding with the element for the given format. Why such a teaching leads the Examiner to believe elements 64 and 66 demarcate a target audience of the document element is not explained in the Office Action. Furthermore, the Examiner's interpretation of Crockett is in error.

Elements 64 and 68 of Figure 6 in Crockett illustrate line elements "Job Name", "Job Step Name" and "Proc Step Name". Crockett describes Figure 6 as follows:

> FIG. 6 illustrates an example of documentation format information 62 that is passed to the documentation checker tool 20. In particular, the documentation format information 62 has on each line 68 an element 64 and text 66. The element or tag indicates an element to be included in the documentation and the text 66 identifies the text that corresponds with the element for the given format. Hence, the job name element has the associated text "Job Name:". Accordingly, each of the desired elements may be identified within the documentation format information 62 to realize documentation like that specified above. The documentation checker tool 20

> uses the documentation format information 62 that is input to it to build a list of tags and text (step 50 in FIG. 5). Crockett, col. 6, ln. 39-52 (emphasis added).

Contrary to the examiner's position, and as discussed in Crockett, Figure 6 shows documentation formatting information, not demarcating a target audience of the document element. In other words, each line 68 has an element 64 to be included in the documentation. There is no mention or suggestion in Crockett of demarcating a target audience of the document element.

In section 10C of the Examiner's Answer, the Examiner asserts, "the documentation format information, e.g., "JOB NAME", "JOB NARRATIVE" which will reference by the documentation checker tools (item 20) to demarcating/identifying the target audience of the document element." EA, pp. 16. The Examiner provides no citation in Crockett supporting this assertion.

By contrast, in describing how documentation format information elements of Fig. 6 are used by the documentation checker tool, Crockett states,

> The documentation is passed into the documentation checker tool (step 56 in FIG. 5). The documentation checker tool uses the documentation format information 62 (FIG. 6) to locate elements within the documentation by identifying the text 66 that is associated with the elements. This is then compared with the lists 70, 72 and 74 that were derived by processing the [Job Control Language] statements to determine whether the documentation is present, complete and accurate (step 58 in FIG. 5). The documentation checker tool performs the comparison for each of the elements that needs to be documented in the JCL instructions. As a result of this

> comparison, the documentation checker tool 20 verifies whether each of the elements is properly documented or not. Crockett, col. 7, ll. 39-51.

The above passage of Crockett indicates the documentation checker tool uses the documentation format information to confirm whether the documentation is present, complete and accurate. Notably absent from Crockett is any disclosure that the documentation checker tool uses the documentation format information to demarcate or identify a target audience of the document element, as advanced by the Examiner. The Examiner's unsubstantiated assertion is therefore clearly in error.

The Appellants submit the rejection of claim 1 is in error and respectfully request that the rejection of claim 1 under 35 U.S.C. § 103 be reversed by the honorable Board.

Claims 2-8

Claims 2-8 are dependent on and further limit claim 1. Since the rejection of claim 1 is believed error for the reasons set forth above, the rejection of claims 2-8 is also believed in error for at least the same reasons as discussed for claim 1.

Claim 14

Claim 14 recites, in part, "at least one document element associated with the source code, the document element including an indicia of a human audience that the document element is targeted to." The Office Action argues this claim element is provided by Crockett. FOA, pp. 11. The Appellants respectfully disagree.

As discussed above for claim 1, Crockett does not teach or suggest a document element including an indicia of a target audience. Figure 6 of Crockett shows formatting information, not demarcating a target audience of the document element. There is no mention or suggestion in Crockett of demarcating a target audience of the document element.

In rejecting claim 14, the Examiner argues Crockett "does not explicitly disclose such audience is human audience." FOA, pp. 11. It is clear the Examiner recognizes Crockett fails to disclose a document element including an indicia of a human audience that a document element is targeted to. The Appellants submit prior art references must teach or suggest all the claim limitations to sustain a *prima facie* case of obviousness under 35 U.S.C. § 103.

The Examiner further argues, "Therefore, it would have been obvious to one having ordinary skill in the art at the time the invention was made to use the element tags to identify the different target human audience, the checker in Crockett can decide where the document element is targeted to according the predefined the template as disclosed (col. 5-6, the example of an appropriate documentation format)." FOA, pp. 11. The Appellants respond that the template disclosed in column 5-6 of Crockett does not mention or provide any suggestion of indicia of a human audience that a document element is targeted to.

The Examiner also directs the reader to Crockett's disclosure that "The documentation checker tool 20 must next process the JCL instructions 24 in a job 22 in order to identify what elements need to be documented in the documentation." FOA, pp. 11. In response, the Appellants observe Crockett describes a system where JCL jobs have corresponding separate documentation. Crockett, col. 1, ll. 38-44. The document checker tool disclosed in Crocket verifies whether each element of the JCL job is properly documented or not. Crockett, col. 7, ll. 46-53. Thus, the citation offered in the Office Action does not teach or suggest indicia of a human audience that a document element is targeted to.

The Appellants therefore submit the rejection of claim 14 is in error and respectfully request that the rejection of claim 14 under 35 U.S.C. § 103 be reversed by the honorable Board.

Claims 15 and 16

Claims 15 and 16 are dependent on and further limit claim 14. Since claim 14 is believed error for the reasons set forth above, the rejection of claims 15 and 16 is also believed in error for at least the same reasons as discussed for claim 14.

Claims 17-19 and 21-24

Claims 17-19 and 21-24 were rejected under the same grounds as claims 1-3, 6-9 and 14-16. Since the rejection of claims 1-3, 6-9 and 14-16 is believed in error for the reasons set forth herein, the rejection of claims 17-19 and 21-24 is also believed in error.

IV. CLAIMS 9 AND 25-29 ARE NOT OBVIOUS OVER CROCKETT IN VIEW OF WINZIP AND FCC

Claims 9 and 25-29 were rejected under substantially the same grounds as claims 1-9. Since the rejection of claims 1-9 is believed in error for the reasons set forth herein, the rejection of claims 9 and 25-29 is also believed in error.

V. CLAIMS 30 AND 31 ARE NOT OBVIOUS OVER CROCKETT IN VIEW OF WINZIP AND HEESCH

Claims 30 and 31 were rejected under substantially the same grounds as claim 1. Since the rejection of claim 1 is believed in error for the reasons set forth above, the rejection of claims 30 and 31 is also believed in error.

Conclusion

In view of the foregoing, Appellant submits that the rejections of Claims 1-9 and 14-21 are improper and respectfully requests that the rejections of Claims 1-9 and 14-21 be reversed by the Board.

Dated: April 10, 2009

Respectfully submitted,
/ido tuchman/
Ido Tuchman, Reg. No. 45,924

Law Office of Ido Tuchman
82-70 Beverly Road
Kew Gardens, NY 11415
Telephone (718) 544-1110
Facsimile (718) 374-6092

Claims Appendix

Claim 1. (previously presented) A method for creating a minimum executable unit, the method comprising:

receiving at least one block of source code;

receiving at least one document element associated with the source code;

demarcating a target audience of the document element;

validating consistency between the source code and the document element; and

building an executable unit, the executable unit being responsive to both the source code and the document element.

Claim 2. (original) The method of claim 1, further comprising receiving at least one external document for inclusion in the executable unit.

Claim 3. (previously presented) The method of claim 1, wherein the document element includes tag-based syntax, the tag-based syntax including the target audience of the document element.

Claim 4. (canceled)

Claim 5. (previously presented) The method of claim 1, further comprising providing a text editor configured to create the document element with tag-based syntax to demarcate information, the tag-based syntax including the target audience of the document element.

Claim 6. (original) The method of claim 1, further comprising extracting the document element from the executable unit in such a way as to be readable as documentation.

Claim 7. (original) The method of claim 1, further comprising aggregating the document element with other document elements.

Claim 8. (original) The method of claim 7, wherein the other document elements are extracted from other executable units.

Claim 9. (original) The method of claim 1, further comprising encrypting the document element.

Claims 10-13 (cancelled)

Claim 14. (previously presented) A system for creating a minimum executable unit, the system comprising:

at least one block of source code;

at least one document element associated with the source code, the document element including an indicia specifying a human audience that the document element is targeted to;

a validation criteria configured to validate the consistency between the source code and the document element; and

a builder configured to construct an executable unit, the executable unit including the document element embedded in.

Claim 15. (original) The system of claim 14, further comprising an import module coupled to the builder and configured to receive document elements from external document providers.

Claim 16. (original) The system of claim 14, further comprising a repository access module coupled to the builder and configured to store and retrieve document elements in at least one database.

Claim 17. (previously presented) A computer program product embodied in a tangible media comprising:

computer readable program codes coupled to the tangible media for creating a minimum executable unit, the computer readable program codes configured to cause the program to:

receive at least one block of source code;

receive at least one document element associated with the source code, the document element including an indicia specifying a human audience that the document element is targeted to;

validate consistency between the source code and the document element; and

build an executable unit, the executable unit configured to be responsive to both the source code and the document element.

Claim 18. (original) The computer program product of claim 17, further comprising program code configured to receive at least one external document for inclusion in the executable unit.

Claim 19. (previously presented) The computer program product of claim 17, wherein the document element includes tag-based syntax to demarcate information, the tag-based syntax including the indicia specifying the human audience that the document element is targeted to.

Claim 20. (canceled)

Claim 21. (previously presented) The computer program product of claim 17, further comprising program code configured to provide a text editor configured to create the document element with tag-based syntax to demarcate information, the tag-based syntax including the indicia specifying the human audience that the document element is targeted to.

Claim 22. (original) The computer program product of claim 17, further comprising program code configured to extract the document element from the executable unit in such a way as to be readable as documentation.

Claim 23. (original) The computer program product of claim 17, further comprising program code configured to aggregate the document element with other document elements.

Claim 24. (original) The computer program product of claim 17, wherein the other document elements are extracted from other executable units.

Claim 25. (previously presented) The computer program product of claim 17, further comprising program code configured to encrypt the document element.

Claim 26. (previously presented) Program code embodied on computer-readable media, the program code comprising:

at least one block of source code providing executable computer instructions; and at least one documentation string coupled to the source code, the documentation string including:

a method indicia configured to identify the method where the documentation string is located;

a target indicia configured to identify who the documentation string is targeted to; and

a granularity indicia configured to identify what document the documentation string should be a part of.

Claim 27. (previously presented) The program code of claim 26, wherein the documentation string further includes an identification indicia, a timestamp, and a sequence number.

Claim 28. (previously presented) The program code of claim 26, wherein the documentation string further includes tag-based syntax to demarcate information.

Claim 29. (previously presented) The program code of claim 26, wherein the documentation string is encrypted.

Claim 30. (previously presented) The method of claim 1, wherein the document element is embedded in the source code.

Claim 31. (previously presented) The computer program product of claim 17, wherein the document element is embedded in the source code.

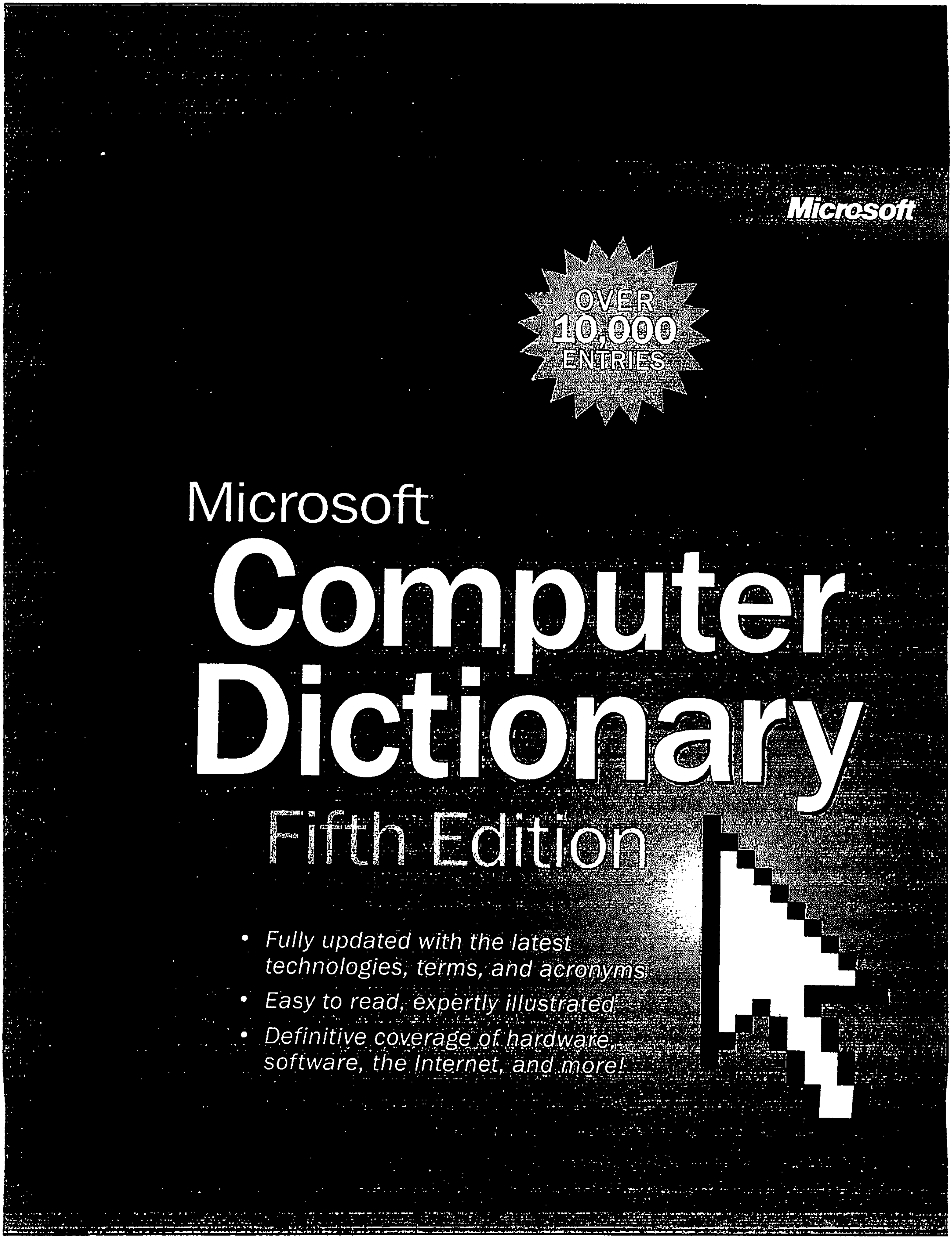
Evidence Appendix

Exhibit A. Microsoft Computer Dictionary Fifth Edition, Microsoft Press, pp. 200 (2002) (definition for executable program).

Related Proceedings Appendix

None.

EXHIBIT A



Microsoft

OVER 10,000 ENTRIES

Microsoft Computer Dictionary

Fifth Edition

- *Fully updated with the latest technologies, terms, and acronyms*
- *Easy to read, expertly illustrated*
- *Definitive coverage of hardware, software, the Internet, and more!*

PUBLISHED BY
Microsoft Press
A Division of Microsoft Corporation
One Microsoft Way
Redmond, Washington 98052-6399

Copyright © 2002 by Microsoft Corporation

All rights reserved. No part of the contents of this book may be reproduced or transmitted in any form or by any means without the written permission of the publisher.

Library of Congress Cataloging-in-Publication Data
Microsoft Computer Dictionary.--5th ed.
p. ; cm.
ISBN 0-7356-1495-4
1. Computers--Dictionaries. 2. Microcomputers--Dictionaries.

AQ76.5. M52267 2002
004'.03--dc21 200219714

Printed and bound in the United States of America.

1 2 3 4 5 6 7 8 9 QWT 7 6 5 4 3 2

Distributed in Canada by Penguin Books Canada Limited.

A CIP catalogue record for this book is available from the British Library.

Microsoft Press books are available through booksellers and distributors worldwide. For further information about international editions, contact your local Microsoft Corporation office or contact Microsoft Press International directly at fax (425) 936-7329. Visit our Web site at www.microsoft.com/mspress. Send comments to *mspinput@microsoft.com.*

Active Desktop, Active Directory, ActiveMovie, ActiveStore, ActiveSync, ActiveX, Authenticode, BackOffice, BizTalk, ClearType, Direct3D, DirectAnimation, DirectDraw, DirectInput, DirectMusic, DirectPlay, DirectShow, DirectSound, DirectX, Entourage, FoxPro, FrontPage, Hotmail, IntelliEye, IntelliMouse, IntelliSense, JScript, MapPoint, Microsoft, Microsoft Press, Mobile Explorer, MS-DOS, MSN, Music Central, NetMeeting, Outlook, PhotoDraw, PowerPoint, SharePoint, UltimateTV, Visio, Visual Basic, Visual C++, Visual FoxPro, Visual InterDev, Visual J++, Visual SourceSafe, Visual Studio, Win32, Win32s, Windows, Windows Media, Windows NT, Xbox are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries. Other product and company names mentioned herein may be the trademarks of their respective owners.

The example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious. No association with any real company, organization, product, domain name, e-mail address, logo, person, place, or event is intended or should be inferred.

Acquisitions Editor: Alex Blanton
Project Editor: Sandra Haynes

Body Part No. X08-41929

Table E.1 Exclusive OR.

a	*b*	*a XOR b*
0	0	0
0	1	1
1	0	1
1	1	0

.exe *n.* In MS-DOS, a filename extension that indicates that a file is an executable program. To run an executable program, the user types the filename without the .exe extension at the prompt and presses Enter. *See also* executable program.

executable[1] *adj.* Of, pertaining to, or being a program file that can be run. Executable files have extensions such as .bat, .com, and .exe.

executable[2] *n.* A program file that can be run, such as file0.bat, file1.exe, or file2.com.

executable program *n.* A program that can be run. The term usually applies to a compiled program translated into machine code in a format that can be loaded into memory and run by a computer's processor. In interpreter languages, an executable program can be source code in the proper format. *See also* code (definition 1), compiler (definition 2), computer program, interpreter, source code.

execute *vb.* To perform an instruction. In programming, execution implies loading the machine code of the program into memory and then performing the instructions.

execute in place *n.* The process of executing code directly from ROM, rather than loading it from RAM first. Executing the code in place, instead of copying the code into RAM for execution, saves system resources. Applications in other file systems, such as on a PC Card storage device, cannot be executed in this way. *Acronym:* XIP.

execution time *n.* The time, measured in clock ticks (pulses of a computer's internal timer), required by a microprocessor to decode and carry out an instruction after it is fetched from memory. *Also called:* E-time. *See also* instruction time.

executive *n.* The set of kernel-mode components that form the base operating system for Microsoft Windows NT or later. *See also* operating system.

executive information system *n.* A set of tools designed to organize information into categories and reports. Because it emphasizes information, an executive information system differs from a decision support system (DSS), which is designed for analysis and decision making. *Acronym:* EIS. *Compare* decision support system.

exerciser *n.* A program that exercises a piece of hardware or software by running it through a large set of operations.

exit *vb.* In a program, to move from the called routine back to the calling routine. A routine can have more than one exit point, thus allowing termination based on various conditions.

expanded *adj.* A font style that sets characters farther apart than the normal spacing. *Compare* condensed.

expanded memory *n.* A type of memory, up to 8 MB, that can be added to IBM PCs. Its use is defined by the Expanded Memory Specification (EMS). Expanded memory is not accessible to programs in MS-DOS, so the Expanded Memory Manager (EMM) maps pages (blocks) of bytes from expanded memory into page frames in accessible memory areas. Expanded memory is not needed in Windows 9x, all versions of Windows NT, and Windows 2000. *See also* EEMS, EMS, Expanded Memory Manager, page frame.

Expanded Memory Manager *n.* A driver that implements the software portion of the Expanded Memory Specification (EMS) to make expanded memory in IBM and compatible PCs accessible. *Acronym:* EMM. *See also* EMS, expanded memory, extended memory.

Expanded Memory Specification *n. See* EMS.

expansion *n.* A way of increasing a computer's capabilities by adding hardware that performs tasks that are not part of the basic system. Expansion is usually achieved by plugging printed circuit boards (expansion boards) into openings (expansion slots) inside the computer. *See also* expansion board, expansion slot, open architecture (definition 2), PC Card, PCMCIA slot.

expansion board *n.* A circuit board that is plugged into a computer's bus (main data transfer path) to add extra functions or resources to the computer. Typical expansion boards add memory, disk drive controllers, video support, parallel and serial ports, and internal modems. For laptops and other portable computers, expansion boards come in credit card-sized devices called PC Cards that plug into a slot in the side or back of the computer. *Also called:* expansion board, extender board. *See also* expansion slot, PC Card, PCMCIA slot.

expansion bus *n.* A group of control lines that provide a buffered interface to devices. These devices can be located